REMARKS

Claims 18-75 had been withdrawn pursuant to an election responsive to the restriction. Claims 1-75 are cancelled without prejudice or disclaimer. New claims 76-98 are added for prosecution at this time. Therefore, claims 76-98 are the claims currently pending in the Application.

Response To Rejections Under 35 U.S.C.§ 112 and 35 U.S.C.§ 102

The Examiner rejected claims 5-17 under 35 U.S.C. § 112, second paragraph. As claims 1-17 have been cancelled, this rejection is moot. It is submitted that new claims 76-98 are fully compliant with 35 U.S.C. § 112, second paragraph.

In addition, the Examiner rejected claims 1-17 under 35 U.S.C. § 102(e) as being anticipated by Fucarile et al., U.S. 6,766,305 B1. The rejection under 35 U.S.C. § 102 is also moot because claims 1-17 have been cancelled. In the following remarks, Applicant sets forth reasons why new claims 75-98 are allowable over Fucarile et al., as well as U.S. Patents 6,345,256 and 6,006,190 both listed on the Examiner's PTO form 892 attached to the Office Action mailed May 6, 2005.

New Claims 76-98

New claims 76-98 introduce no impermissible new subject matter. Independent claim 76 is fully supported by Applicant's disclosure, see for example, page 32, last paragraph; page 33; and original claims 6, 10, 13, 14 and 17. The new independent claim 76 is now directed to a method of modifying an executable file, a method which is generally described in the original specification beginning on page 24, third paragraph. It is to be noted that for a more concise description of the claimed

subject matter, the term "subroutine" used in the originally filed claim set has now been replaced by the expression "instruction."

It is believed that the new claims 76 to 98 now particularly point out and distinctly claim the subject matter, which the applicant regards as the invention. Further, it is believed that the subject matter of the new claims is patentable over the prior art as set out below.

Formal Matters

Applicant thanks the Examiner for acknowledging the claim for foreign priority and the receipt of the priority document.

Technical Overview

Generally, the present invention relates to the field of digital rights management (DRM) to ensure that a user is able to use content only in compliance with the licensing conditions under which the content was acquired. For example, playing a song five times or using a piece of software on a single designated computer may be allowed.

In general, content usage is controlled by a usage monitoring mechanism that "knows" about the licensing conditions that apply to given content and which either grants or denies a certain kind of usage. When the user tries to play a song, the usage monitoring mechanism is consulted and, if the song has not yet been played five times, the usage monitoring mechanism will grant the request to play the song. Similarly, the request to run the piece of software would be granted if it is detected that the piece of software is run on the legitimate designated computer.

The state of the art in DRM is characterized by the struggle to develop

attack-resistant methods that address the following two challenges, namely:

1. to tell the usage monitoring mechanism which licensing conditions apply to given content, and

2. prevent the user from accessing content without control by the usage monitoring mechanism.

The first challenge is typically addressed by creating a license descriptor that specifies the applicable rights for given content. The license descriptor is protected against illicit modification by digital signatures and is linked to the given content by means of cryptographically secure hash functions. Thus, the license descriptor cannot be modified or attached to different content.

The second challenge is typically addressed by encrypting the content, preferably employing public key cryptography and embedding functionality for the description of content in the usage monitoring mechanism. Hence, only the usage monitoring mechanism is able to gain access to unencrypted content. The usage monitoring mechanism will, however, always honor the license descriptor that accompanies the content.

Thus, content usage is tied to the usage monitoring mechanism and, consequently, to the rights granted by the license descriptor. Stated simply, the state of the art knows very well which functionality the usage monitoring mechanism has to implement and which data formats ("secured containers") suit this functionality best. Apart from these two solved questions the security of DRM solution depends on two qualities of the usage monitoring mechanism, namely:

1. the usage monitoring mechanism prevents the interception of decrypted

content during usage, and

2. the usage monitoring mechanism is tamper-resistant.

If it is easy for a user to intercept the decrypted representation of a song while it is played, the user is able to store the song in a format which is DRM agnostic, e.g. MP3, and use it without any restrictions.

If it is easy for a user to reverse engineer and to modify the usage monitoring mechanism, the user is able to obtain grant for usage requests unconditionally irrespective of the license descriptor and thus to use the content again and again without any restrictions.

In the prior art, there is a strict separation between the piece of software that implements the usage monitoring mechanism and the content, which has proven to be a big disadvantage.

US 6,817,535 B2 for example suggests embedding the usage monitoring mechanism into "conforming devices" (column 1, lines 60). The conforming device then operates on the content stored on a recording medium. Here the separation is obvious and cannot be overcome. The usage monitoring mechanism is a piece of software that consists of executable instructions, whereas the content is strictly passive data that cannot be executed but that can only be operated on. It is impossible to mix program code instructions and data. US 6,006,190 adds the usage monitoring mechanism to the executable file to be protected. However, the separation is maintained.

The usage monitoring mechanism consists of a "shell" which is fused around the original code (column 3, lines 15 to 16). In this case, however, the separation between the usage monitoring mechanism and the content is not inherent but due to the

limitations of the state of the art. The usage monitoring mechanism is a piece of software that consists of executable instructions. The content is also a piece of software that consists of executable instructions, but the separation is maintained. The content forms an individual monolithic block, e.g. main install program" (column 8, line 61) and the usage monitoring mechanism consists of two further monolithic blocks, e.g. "header part of the install shell program" (column 8, line 54) and "install middle part" (column 8, line 58).

Rejection under the Cited Art

For at least the following reasons, Applicant's claimed invention is neither anticipated by nor obvious from the cited reference. Independent claims 76 and 98 require embedding data and/or code before or after the identified instructions, whereby the embedded code is formed by a license verification code and the embedded data represents license related information.

US 6,766,305 B1 also maintains this separation between the usage monitoring mechanism and the content. Program code is modified, but only in order to inject the license descriptor "license form" (column 3, line 20), which specifies the rights to be granted by the usage monitor. The usage monitoring mechanism, however, is a monolithic block completely separate from the content, implemented by a "plug-in or application" (column 3, lines 34 to 35). The usage monitoring mechanism "scans the content for the license form" (column 3, line 32), i.e. it searches content for a license descriptor and then enforces the specified license conditions. It allows operation "on the content to the extend allowed by the access policy" (column 3, lines 32 to 33).

US 6,345,256 B1 also maintains this separation, as can be seen in the

section "X. End-User Devices" (column 82, line 12) of the document. Here, the usage monitoring mechanism consists of an "executable application which is configured as a helper application into the end-user(s) web browser" (column 82, lines 34 to 35). In this context, the usage monitoring mechanism is named "SC(s) processor" (column 82, line 32). The SC(s) processor, e.g., decrypts received audio content, watermarks it and re-encrypts it for storage on the user's playback device.

Both, US 6,006,190 and US 6,345,256 B1 explain how the current state of the art tries to create tamper-resistant usage monitoring mechanisms. The author of tamper-resistant codes adds functionality to the code that tries to detect if an attacker attaches a debugger to the running code and that keeps large portions of the code encrypted until it is executed, then decrypts the code and, after execution, reencrypts it (US 6,006,190 in column 84, lines 8 to 18 and US 6,345,256 B1 in column 5, line 42 to column 6 line 4). Thus, the prior art in tamper-resistant code generally has two functionalities, namely:

1. to detect debuggers and refuse to run in the presence of a debugger; and
2. to decrypt code only when it is about to be executed and re-encrypt afterwards.

Independent claims 76 and 98 require embedding data and/or code before or after the identified instructions, whereby the embedded code is formed by a license verification code and the embedded data represents license related information.

According to an aspect of Applicant's claimed invention, modification of an executable file is provided, an executable file is secured against software piracy in a simple and very effective manner. Specifically, the present invention ensures that only

legitimate users can run the executable file and/or that users can use the executable file only in compliance with the applicable licensing terms. Specifically, the present invention suggests to link (embed) license verification code and/or license verification information to the executable file in a way that makes it impossible to unlink it again so that the executable file is fully protected.

According to an aspect of the present invention, insertion of the instructions of the usage monitoring code (license verification code and license related information) between the instructions of the original code is enabled by embedding, so that after re-assembly of the executable file, a single monolithic block containing the original instructions as well as the usage monitoring instructions is obtained. Accordingly, the present invention enables for the first time automatically embedding pieces of usage monitoring code at a large number of locations such that manual removal by a human becomes infeasible because of the involved workload.

It is to be noted that US 6,006,190 suggests uniting the usage monitoring mechanism and the original code in a single execution file. However, the approach of this prior art document is to handle the original code and the usage monitoring code as two separate blocks, so that it is relatively easy for an attacker to identify and separate the monolithic block of original code and the monolithic block of usage monitoring code in order to undo the transformations and to remove the usage monitoring code. The other prior art documents do not even suggest combining the usage monitoring code arid the content into a single file.

It is therefore believed that the subject matter of the present invention is not only novel but also nonobvious over the prior art.

For at least the reasons set forth in the foregoing discussion, Applicant believes that the Application is now allowable, and respectfully requests that the Examiner reconsider the rejections and allow the Application. Should the Examiner have any questions regarding this Amendment, or regarding the Application generally, the Examiner is invited to telephone the undersigned attorney.

Respectfully submitted,

Paul J. Esatto, Jr.
Registration No. 30,749

Scully, Scott, Murphy & Presser, P.C.
400 Garden City Plaza, Suite 300
Garden City, New York 11530
(516) 742-4332
PJE:GB:ar:ae